UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/663,758	09/17/2003	William Patrick Tunney	11884/404001	3826

23838 7590 07/25/2006

KENYON & KENYON LLP
1500 K STREET N.W.
SUITE 700
WASHINGTON, DC 20005

EXAMINER	
ZHEN, LI B	

ART UNIT	PAPER NUMBER
2194	

DATE MAILED: 07/25/2006

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/663,758	TUNNEY, WILLIAM PATRICK	
	Examiner	Art Unit	
	Li B. Zhen	2194	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *17 September 2003*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-27* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-27* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☒ The drawing(s) filed on *17 September 2003* is/are: a)☒ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date *11/13/03*.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-27 are pending in the application.

Information Disclosure Statement

2. The information disclosure statement (IDS) submitted on 11/13/2003 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is has been considered by the examiner.

Specification

3. The title of the invention is not descriptive. A new title is required that is clearly indicative of the invention to which the claims are directed.

Claim Objections

4. Claim 7 is objected to because of the following informalities: the abbreviations (SQL) in the claim should be defined. Appropriate correction is required.

Claim Rejections - 35 USC § 112

5. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

6. Claim 21 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

7. Claim 21 recites the limitation " the message iterator object " in line 1. There is insufficient antecedent basis for this limitation in the claim.

Claim Rejections - 35 USC § 101

8. 35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

9. Claims 1-18 and 25-27 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter. The claimed invention as a whole must be useful and accomplish a practical application. That is, it must produce a "useful, concrete and tangible result." State Street, 149 F.3d at 1373-74, 47 USPQ2d at 1601-02. Examiner suggests that independent claims 1, 8, 10 and 25 do not appear to provide a "useful, concrete and tangible result". The result of independent claims 1, 8 and 10 are "iterating" through data element or message. Iterate through data or message means to access/read the elements within the data or message. Claims 1, 8 and 10 do not require any operation on the data element after accessing/reading the element. Examiner suggests accessing or reading elements without performing any operation with the accessed data does not produce a tangible result; therefore, claims 1, 8 and 10 are non-statutory. Claim 25 produces a result of performing an operation on the element object. The claim does not specifically recite what type of operation is performed. A reasonable interpretation, in view of independent claims 1, 8 and 10, would be the operation is an iteration operation. Under this interpretation, claim 25 also

results in iterating through an element object. Therefore, claim 25 also does not produce a tangible result for the same reasons as claims 1, 8 and 10.

Claim Rejections - 35 USC § 102

10. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

11. Claims 1-4, 7-9, 19-21 and 24-27 are rejected under 35 U.S.C. 102(b) as being anticipated by U.S. Patent No. 6,421,661 to Doan et al. [hereinafter Doan].

12. As to claim 1, Doan teaches a method comprising:

receiving a call [search criteria] to iterate through a collection [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7] including at least one uninstantiated element [a collection of BOs 206 and/or DOs 208; col. 10, lines 5 – 16];

after receiving the call, instantiating the uninstantiated element [BOs 206 and their corresponding DOs 208] to provide an instantiated element [iterator object 210 instantiates the BOs 206 and their corresponding DOs 208 during execution; col. 8, lines 42 - 56]; and

iterating through the instantiated element [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208 to materialize one BO 206 and/or DO 208; col. 10, lines 5 – 16].

13. As to claim 8, Doan teaches a method comprising:

after receiving a call [search criteria] to iterate through raw data [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7; examiner interprets raw data as unprocessed data and collection of BOs 206 and Dos 208 correspond to raw data because the BOs 206 and DOs 208 are instantiated just prior to iteration, col. 10, lines 5 – 16. After instantiation, the BOs 206 and DOs 208 would contain default values that have not been processed], generating data elements [BOs 206 and their corresponding DOs 208] from the raw data [iterator object 210 instantiates the BOs 206 and their corresponding DOs 208 during execution; col. 8, lines 42 - 56], wherein the data elements can be iterated through [col. 10, lines 5 – 16]; and

iterating through the data elements [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208 to materialize one BO 206 and/or DO 208; col. 10, lines 5 – 16].

14. As to claim 19, Doan teaches a machine readable medium [col. 7, lines 22 – 33] containing program instructions for execution on a processor [col. 7, lines 22 – 33], which when executed by the processor, cause the processor to perform:

calling a message reader object [objects framework 108 receives IMS.TM. transaction requests from a requestor; col. 8, lines 42 – 56] to iterate through a collection [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7];

creating a file message reader object [application program 106 then creates the iterator object 210; col. 9, line 65 – col. 10, line 7] to determine the format [DL/I.TM. query string comprises a "data type string"; col. 7, lines 5 – 18] of an element in the collection [creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7];

using the file message reader object to read the element [DOs 208] from the collection [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208 to materialize one BO 206 and/or DO 208; col. 10, lines 5 – 16]; and

after the calling the message reader object [objects framework 108 receives IMS.TM. transaction requests from a requestor via one or more instantiated message queue objects 212; col. 8, lines 43 – 56], using the file message reader object to create [After execution, responses are returned to the requestor as IMS.TM. transaction

responses via the instantiated message queue objects 212; col. 8, lines 42 – 56] a message object [message queue objects; col. 13, lines 26 – 33] from the read element [responses].

15. As to claim 25, Doan teaches a method comprising:

responsive to a request [search criteria] to read an element from a collection [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7], creating a reader object [iterator object] based on the format [DL/I.TM. query string comprises a "data type string"; col. 7, lines 5 – 18] of the element [creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7];

opening the collection [objects framework instantiates IMS.TM. data objects upon demand from application programs and manages those objects from creation to deletion; col. 6, lines 43 – 56];

after receiving the request [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7], using the reader object [iterator object210], instantiating an element object including the element [iterator object 210 instantiates the BOs 206 and

their corresponding DOs 208 during execution; col. 8, lines 42 - 56], and performing an operation on the element object [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208 to materialize one BO 206 and/or DO 208; col. 10, lines 5 – 16]; and

closing the collection [After execution, responses are returned to the requestor; col. 8, lines 42 – 56; manages those objects from creation to deletion, col. 6, lines 43 - 56].

16. As to claim 2, Doan teaches implementing an interface [col. 8, lines 32 – 43] having routines ["next" method of the iterator object 210 is invoked to instantiate each BO 206 and/or DO 208; col. 10, lines 6 - 16] for iterating through the collection [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208; col. 10, lines 6 - 16].

17. As to claim 3, Doan teaches implementing an interface [col. 8, lines 32 – 43] having routines ["next" method of the iterator object 210 is invoked to instantiate each BO 206 and/or DO 208; col. 10, lines 6 - 16] for instantiating the uninstantiated element based on the format [data types; col. 6, line 65 – col. 7, line 17] of the uninstantiated element [invoking the "next" member function or method of the iterator object 210 to instantiate/materialize a DO 208 and/or BO 206 in the memory; col. 14, lines 22 - 28].

18. As to claim 4, Doan teaches determining whether the uninstantiated element is available in the collection [iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 9].

19. As to claim 7, Doan teaches the collection is an SQL [col. 7, lines 5 – 23] database [col. 7, lines 38 – 65] table [records in the database 112... collection of data objects (DOs) 208 and business objects (BOs) 206; col. 9, 15 – 26; examiner notes that a table is a collection of records in a database] and the uninstantiated element is a database field [DO 208; col. 14, lines 60 – 67].

20. As to claim 9, Doan teaches the data elements are generated based on the format [data types; col. 6, line 65 – col. 7, line 17] of the raw data [invoking the "next" member function or method of the iterator object 210 to instantiate/materialize a DO 208 and/or BO 206 in the memory; col. 14, lines 22 – 28 and col. 9, line 65 – col. 10, line 7].

21. As to claim 20, Doan teaches the file message reader object includes a routine to retrieve the element [Block 414 represents the application program 106 retrieving data from the database 112 via a method of the DO 208; col. 14, lines 25 – 35] from the collection and create the message object [message queue objects; col. 13, lines 26 – 33] from the retrieved element [After execution, responses are returned to the requestor

as IMS.TM. transaction responses via the instantiated message queue objects 212; col. 8, lines 42 – 56].

22. As to claim 21, Doan teaches the message iterator object includes a first routine to create [objects framework 108 also dynamically loads the class library for the BOs 206 and DOs 208 requested by the application program 106 to create an iterator object 210; col. 8, lines 42 – 56] the file message reader object [iterator object 210] and a second routine to iterate through the collection [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208; col. 10, lines 5 – 16].

23. As to claim 24, Doan teaches determining the format of the element during the execution by reading at least a portion of the element and evaluating the portion [A DO 208 class represents each segment type and each segment occurrence is represented by an instance of the class, i.e., a DO 208; col. 9, lines 26 – 40].

24. As to claim 26, Doan teaches repeating the instantiating the element object and performing if there are additional elements available to be read from the collection [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208 to materialize one BO 206 and/or DO 208; col. 10, lines 5 – 16].

25. As to claim 27, Doan teaches removing the element from the collection after the instantiating or the performing [deleting data from the database 112 via a method of the DO 208; col. 14, lines 40 – 49].

Claim Rejections - 35 USC § 103

26. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

27. Claims 5, 10-18, 22 and 23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Doan in view of U.S. Patent No. 6,820,081 to Kawai et al. [hereinafter Kawai].

28. As to claim 10, Doan teaches the invention substantially as claimed including a method comprising:

receiving an instruction to iterate [search criteria] through a file [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7; examiner notes that a file is a collection of data and Doan discloses a collection of business object and data objects; therefore, the collection of business and data objects corresponds to

the recited file] including at least one data element [a collection of BOs 206 and/or DOs 208; col. 10, lines 5 – 16];

determining to which of a plurality of predefined formats [DL/I.TM. query string comprises a "data type string"; col. 7, lines 5 – 18] the data element belongs [creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7];

instantiating the data element [iterator object 210 instantiates the BOs 206 and their corresponding DOs 208 during execution; col. 8, lines 42 - 56] using a routine associated with the determined format of the data element [creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 7]; and

iterating through the instantiated data element [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208 to materialize one BO 206 and/or DO 208; col. 10, lines 5 – 16].

Although Doan teaches message objects [message objects 212 are capable of supporting multiple message segments; col. 10, lines 60 – 67], Doan does not disclose storing the message as a part of the collection of data objects and iterating through the message.

However, Kawai teaches a message processing system [col. 5, lines 22 – 35] including iterating through a file [message stored in an archived message store 236 is processed during an iteration of the inner processing loop (block 322); col. 16, lines 38 – 64] including at least one message [messages in message stores 236; col. 16, lines 38 - 63], creating the message [parsed metadata and message body are then stored in message records 248 maintained in the database 235; col. 14, lines 12 – 27] using a routine associated with the determined format of the data element [converting the message store from a compressed or archival storage format into a standardized "working" message store format; col. 13, line 55 – col. 14, line 15], and iterating through the instantiated message [iteratively process each of the extracted message stores 236 and individual messages to populate the message records 239 stored in the database 235; col. 16, lines 27 – 40].

It would have been obvious to a person of ordinary skilled in the art at the time the invention was made to combine Kawai and Doan because Kawai's teachings provide the benefits of a normalized electronic storage structure of messages that is independent of physical storage media [col. 13, line 55 – col. 14, line 13 of Kawai] and a system that efficiently processes messages through iteration [col. 9, lines 43 – 50 of Kawai].

29. As to claim 5, Doan teaches the collection is a file [application program 106 then creates the iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the

DL/I.TM. query string; col. 9, line 65 – col. 10, line 7; examiner notes that a file is a collection of data and Doan discloses a collection of business object and data objects; therefore, the collection of business and data objects corresponds to the recited file], but does not specify the uninstantiated element is a file message.

However, Kawai teaches a message processing system [col. 5, lines 22 – 35] including iterating through a file [message stored in an archived message store 236 is processed during an iteration of the inner processing loop (block 322); col. 16, lines 38 – 64] including at least one message [message stores 236; col. 16, lines 38 - 63], and iterating through the instantiated message [iteratively process each of the extracted message stores 236 and individual messages to populate the message records 239 stored in the database 235; col. 16, lines 27 – 40]. As to the motivation for combining Doan and Kawai to incorporate the features of storing messages in the file, see the rejection to claim 10 above.

30. As to claim 11, Doan as modified teaches determining whether the file includes subsequent messages [col. 14, lines 12 – 27 of Kawai] to be retrieved ["next" method of the iterator object 210 is invoked to instantiate each BO 206 and/or DO 208; col. 10, lines 6 – 16 of Doan]; and if so, retrieving a next message [Using the pointer and "next" method of the iterator object 202, the application program 106 can iterate through a collection of BOs 206 and/or DOs 208; col. 10, lines 6 – 16 of Doan] using a routine associated with the determined format [DL/I.TM. query string comprises a "data type string"; col. 7, lines 5 – 18 of Doan], and optionally, removing the next message from the

file [deleting data from the database 112 via a method of the DO 208; col. 14, lines 40 – 49 of Doan]. As to the motivation for combining Kawai and Doan to incorporate the features of storing messages in the file, see the rejection to claim 10 above.

31. As to claim 12, Doan as modified teaches opening the file [objects framework instantiates IMS.TM. data objects upon demand from application programs and manages those objects from creation to deletion; col. 6, lines 43 – 56 of Doan]; determining whether the message [col. 14, lines 12 – 27 of Kawai] is available from the file [iterator object 210 that is used to point to an incrementally-materialized collection of BOs 206 and DOs 208 that meet the search criteria specified in the DL/I.TM. query string; col. 9, line 65 – col. 10, line 9 of Doan]; and closing the file after the message has been retrieved [After execution, responses are returned to the requestor; col. 8, lines 42 – 56; manages those objects from creation to deletion, col. 6, lines 43 - 56]. As to the motivation for combining Kawai and Doan to incorporate the features of storing messages in the file, see the rejection to claim 10 above.

32. As to claim 13, Doan teaches the predefined formats are transparent to a user [supports a complete set of data types...allows easy expansion to new data types to be defined; col. 6, lines 55 – 65; examiner notes that new data types could be added, therefore, the user would not know all the predefined formats].

33. As to claim 14, Doan teaches the predefined formats include data bytes [col. 16 – 17, see Table 1, integer (4 bytes), short (2 bytes)]. Doan does not specifically disclose a fixed length header.

However, Kawai teaches predefined formats include a fixed length header [message header 75, 77, 81; col. 9, lines 20 – 30; examiner notes that each header has a maximum length which would define a fixed length] and data bytes [a message body 74, 78, 82; col. 9, lines 20 – 30].

It would have been obvious to a person of ordinary skilled in the art at the time the invention was made to combine Kawai and Doan because Kawai's teachings provides the benefits of storing information about the message at the beginning of the message [col. 7, lines 50 – 67 of Kawai]. In addition, the header information in combination with the body of the message can be used to calculate hash codes for the message [col. 14, lines 26 – 42 of Kawai] and the hash codes allows the messages to be concatenated and stored as a compound document [col. 16, line 63 – col. 17, line 12 of Kawai].

34. As to claim 15, Doan does not teach reading the fixed length header from the file, calculating an integer value of the fixed length header, reading the data bytes disposed in the file after the fixed length header, the number of read data bytes corresponding to the integer value, and returning the read data bytes as the message.

However, Kawai teaches reading the fixed length header from the file [header and a message body are extracted from each of a plurality of messages; col. 3, lines 60

– 67; parser 243 parses individual fields from each extracted message and identifies message routing, identification information and literal content within each field; col. 14, lines 12 – 27];

calculating an integer value of the fixed length header [messages are sorted in descending order of message body length; col. 17, lines 40 – 63];

reading the data bytes disposed in the file after the fixed length header [parsed metadata and message body are then stored in message records 248; col. 14, lines 13 – 27], the number of read data bytes corresponding to the integer value [length 280; col. 15, lines 30 – 45]; and

returning the read data bytes as the message [Each message record 248 includes a hash code 249 associated with the message; col. 14, lines 13 – 27].

It would have been obvious to a person of ordinary skilled in the art at the time the invention was made to combine the teachings of Doan and Kawai because calculation of message length allows messages to be sorted in descending or ascending order [col. 17, lines 40 – 63 of Kawai] and provides a method for identifying and categorizing messages extracted from archived message stores [col. 3, lines 22 – 26 of Kawai].

35. As to claim 16, Doan does not teach the predefined formats include delimiters separating data bytes.

However, Kawai teaches the predefined formats include delimiters separating data bytes [demarcation between the data constituting a header and the data constituting a message body; col. 14, lines 25 – 42].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine Doan and Kawai because the delimiters create separators that can be searched, identified and analyzed in determining duplicate messages [col. 9, lines 30 – 45 of Kawai].

36. As to claim 17, Doan as modified teaches reading the data bytes until a delimiter is reached [col. 9, lines 30 – 45 of Kawai] and returning the read data bytes as the message [parser 243 parses individual fields from each extracted message and identifies message routing, identification information and literal content within each field; col. 14, lines 12 – 27 of Kawai]. As to motivation for combining Doan and Kawai, see the rejection to claim 16 above.

37. As to claim 18, Doan teaches receiving an instruction to iterate through an empty file ["evaluate" method of the applView object 202 reads the DL/I.TM. query string and sets a pointer in the iterator object 210 to point to the collection of BOs 206 and DOs 208 that meet the DL/I.TM. segment search criteria; col. 9, line 65 – col. 10, line 7, examiner notes that when none of the BOs meet the criteria, the collection of BOs and DOs would be empty]; and returning an indication that the empty file does not include any messages [null pointer is returned if processing of the query is not successful; col.

11, lines 15 – 25; examiner notes that when none of the BOs and DOs meet the criteria, the collection would be empty and the result of the query string would be a null pointer, which signifies an empty collection].

38. As to claim 22, see the rejection to claim 14 above.

39. As to claim 23, see the rejection to claim 16 above.

40. Claim 6 is rejected under 35 U.S.C. 103(a) as being unpatentable over Doan in view of U.S. Patent No. 6,438,539 to Korolev et al. [hereinafter Korolev].

41. As to claim 6, Doan does not specifically disclose the collection is a web page and the uninstantiated element is a web link.

However, Korolev teaches a system for searching an information network in response to a specified criterion provided by a network user [col. 2, lines 21 – 27], a web page [Source package 42 represents a data repository for storing data, e.g., Web pages; col. 7, lines 50 – 63], a web link [URL list; col. 14, lines 33 – 49], and iterating through the web link [search agent 102 iterates through the URL list by connecting to the Web servers on the URLs in the list; col. 14, lines 33 - 50].

It would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine Korolev and Doan because Korolev's teachings provides the benefits of providing an information network user with information

contained within at least one of a plurality of network sites in view of a user's search criteria [col. 2, lines 26 – 45 of Korolev] and increases search efficiency of search methods without knowledge of any previously conducted searches related to a similar criteria [col. 2, lines 45 – 56 of Korolev].

Conclusion

42. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

US PATENT NO. 6539398 discloses an object-oriented method programming model for accessing both hierarchical and relational databases from an objects framework.

US PATENT NO. 6539397 discloses a computerized object-oriented method for performing system service requests by modeling system service calls into an object framework.

US PATENT NO. 6529914 discloses an object-oriented programming model for accessing hierarchical databases.

US PATENT NO. 7058620 discloses a system for cross-platform subselect metadata extraction.

US PATENT NO. 6360229 discloses a generic execution model for isolating application programs from the structure and operations of the underlying data.

US PATENT NO. 6202069 discloses a computerized method for accessing hierarchical data using an object-oriented framework.

US PATENT NO. 6128619 discloses generating an Internet application for accessing a hierarchical database using an object-oriented framework.

US PATENT NO. 6128611 discloses an Internet-enabled generic application program for accessing hierarchical data using an object-oriented framework.

US PATENT NO. 6430571 discloses a multi-frame output form that facilitates Internet search and update for hierarchical databases.

US PATENT NO. 6192369 discloses a object-oriented method for accessing transactional requests for a database by modeling I/O message queues into an object framework.

US PATENT NO. 6141660 discloses a method for generating an application for accessing a hierarchical database using an object-oriented framework.

CONTACT INFORMATION

43. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Li B. Zhen whose telephone number is (571) 272-3768. The examiner can normally be reached on Mon - Fri, 8:30am - 5pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, William Thomson can be reached on 571-272-3718. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Li B. Zhen
Examiner
Art Unit 2194

LBZ